

RMarkdown Tutorial for EBIO 3080

YOUR NAME HERE

20 August, 2016

Read through this document carefully! There is an assignment at the end of the tutorial that you will have to submit for credit. If you need help, do not hesitate to ask a TA, Dr. Martin, or your fellow classmates!

This document is an introduction to using RMarkdown, a program for authoring R documents that combines plain text, code, code output, math, and graphics. We will use RMarkdown to create PDFs for lab assignments. Although it takes a little time to learn initially, you will find it is *much* easier to produce high quality documents in this setting compared to using typical word processors.

Instructions

1. Install R

Before anything else make sure you have already installed R. If you have not installed R, go to [R's webpage](#), and download the version that is specific to your computer's operating system (i.e. Windows, Macintosh, or Linux).

2. Install RStudio

RMarkdown is intended to be used within Rstudio. If you do not have RStudio, go to [RStudio's webpage](#), and download the version that is specific to your computer's operating system (i.e. Windows, Macintosh, or Linux). Once installed, reopen this document in RStudio.

3. Install L^AT_EX

In addition to having R and RStudio, you will need to install a version of the program, L^AT_EX. It will take a little more time to install (10-20 minutes), so be patient. Depending on your operating system, install one (*and only one*) of the following:

- For Mac users, please install [Mactex](#). The website offers a full version and a smaller basic version – you MUST download the full version.
- For Windows users, please install [Miktex](#)
- For Linux users, please install [Texlive](#)

4. Install the rmarkdown package

One last thing. You also need to install the RMarkdown package. To install, click the “Tools” tab at the top of your RStudio window. From the drop down menu, choose the “Install Packages...” option. A new window should appear with an empty cell to type into. Search for “rmarkdown”. Make sure the “install dependencies” button near the bottom of the window is checked. Then hit the “install” button. Once this package is installed, you should be all set!

Try “knitting” your document

Before you continue reading, try turning this document – as is – into a PDF. To do so, use the “Knit PDF” button located near the center and top of the RStudio window (the button should have a little ball of yarn with two knitting needles). Once you hit the button, a PDF document should automatically open on your screen. Congratulations, you have created a PDF document using RMarkdown! Spend a little time looking it over, and see how the plain text that you are reading in RStudio connects to the PDF that it generated. If this process did not work, but you have already installed all the software described above, get help from your class TA. We should be able to diagnose the problem and get you back on track.

Combine your thoughts with your code and graphs

Notice that in RStudio, this document is a text file – you can edit any and all of the lines. Be careful though, as you can not have code errors in the document or you will not be able to “knit” it together.

If you are reading this within RStudio the lines directly below the next paragraph should have a light gray background. This is an example of a R code block. Since an RMarkdown document combines text and code, you have to tell the computer which parts of the document should be interpreted as plain text, and which parts should be treated as R code. All R code must be placed in a code block to be interpreted as such. *We will always provide you with code blocks whenever we ask you to create any code in R.* Just like in a regular R script, you can run particular lines of code by hitting the “Control” then “Enter” keys together. This allows you to use R interactively, just how you would outside of RMarkdown, but once you are happy with the result, you do not have to transfer anything to a new document – everything is already bundled right here!

To get our feet wet with using R within RMarkdown, we will use the famous “iris” dataset that comes installed with R. R comes with a TON of datasets intended for researchers to test out their ideas on. To learn more check out the [R data sets website](#). If you want to learn more about the iris dataset, use the “?” function as shown in the code below.

To reemphasize, you will never have to make your own code blocks, we will always provide you with them, but you will frequently have to write your own code within the block. A basic R code block in RMarkdown looks like this:

```
#Comments and code work just like normal R.

#code to make and add two vectors
x <- c(1,2,3)
w <- c(4,5,6)
v <- x + w

#To learn more about the iris dataset,
#place your curser on the code line below and hit
#the Control then Enter keys together
?iris
```

Now we will make a plot with the iris data to get a feel for how graphics look when knitted into a PDF.

```
#Demonstrate plotting with the iris package.

#Notice the use of the the $ operator,
#which allows you to select columns within a dataframe.

plot(iris$Sepal.Length, iris$Sepal.Width, main="Simple R plot",
      xlab="Sepal Length", ylab="Sepal Width")
```

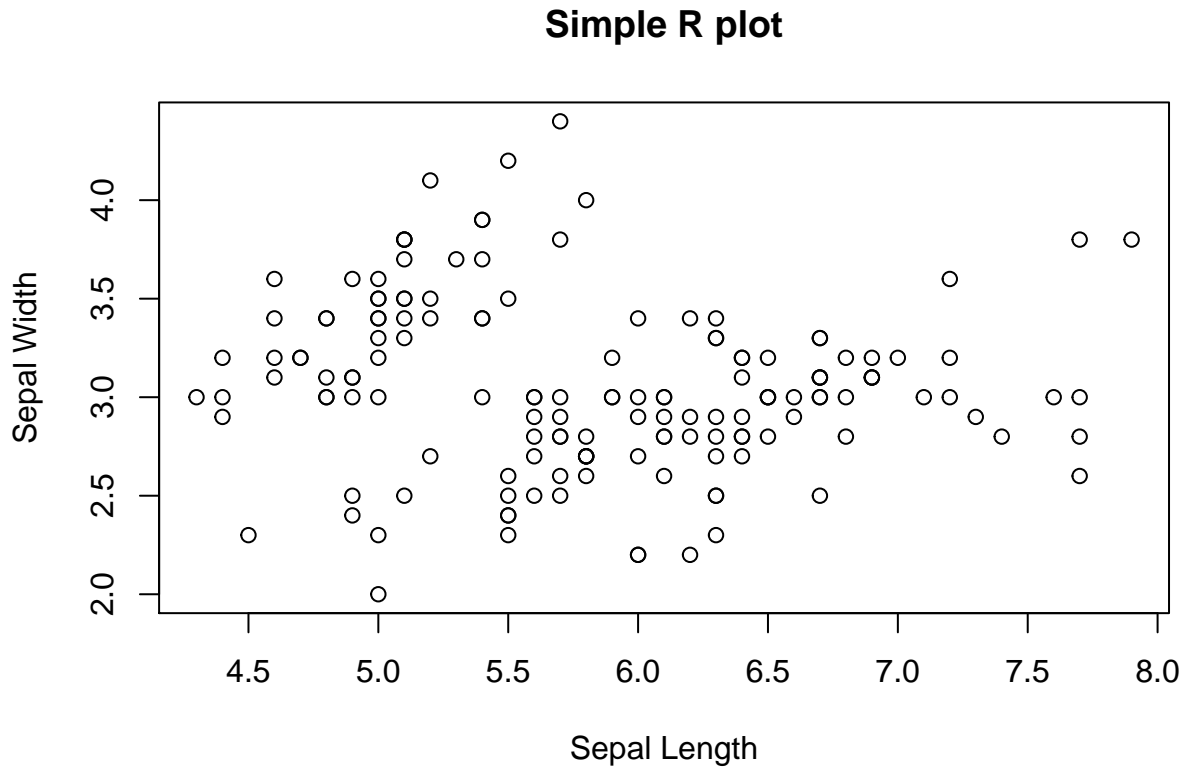


Figure 1: It's easy to create figure captions. Add `'fig.cap ='` to your code block followed by your caption in quotes.

Printing variables inline

A fun trick you can do is to print R variables directly in your text by enclosing some code with back ticks, which is at the top left corner of your keyboard, and the letter “r”, which lets Rstudio know to treat the text as r code. For example the vector “v” in the first code block is equal to: 5, 7, 9. When you “knit” this document, you should see values where the letters ‘r v’ are written in RStudio. This is essentially a mini code block, which is really handy when you need to report values within a sentence. If this does not make sense, don't worry. It's not essential, but just a fun extra.

Assignment

Hopefully this tutorial gave you a good sense of using RMarkdown. You will be surprised at how quickly you improve at using it, and at how awesome your documents will look when you are all done making them. To give you some more practice, please complete the questions below and turn them in *AS A KNITTED PDF* to the D2L assignment dropbox under “RMarkdown Tutorial”.

1. The top eight lines in this document are called the yaml front matter. The front matter provides instructions to your software for how to knit the document. How and why the front matter sections works is not important to us except for the third line. The third line reads:
`'author: "YOUR NAME HERE"'`
Replace “YOUR NAME HERE” with your name (i.e. Marlon Brando). Be sure to leave the double quotes surrounding your name, or your PDF will not knit.

2. Look at the documentation for the iris dataset (hint: line 59). Use the code block below to compute the mean, variance, and standard deviation of the petal width variable. Take some time to mentally reflect on what information each of the three summary statistics convey.
3. Use the iris dataset to make your own summary plot of the data. This could be a histogram of a single variable, bivariate scatter plot, barplot, or anything else. Alternatively, copy the code from the above plot and try making it “prettier” by adding colors, changing point types, etc. Make sure you axes are appropriately labeled. Try adding your own figure caption to your plot (hint: line 65).
4. In a few sentences summarize your general mood about this tutorial and using RMarkdown – say if it was extremely challenging, or if you are excited to use it, etc. Don’t be afraid to tell us how you really feel (within reason).

DELETE THIS SENTENCE AND INSERT YOUR TEXT HERE – KEEP THE STARS, THEY MAKE TEXT ITALIC, WHICH MAKES GRADING EASIER